Linear Regression Sand Box and Open Hand Problem

θ	τ
0.698132	0.188224
0.959931	0.209138
1.134464	0.230052
1.570796	0.250965
1.919862	0.313707

Indexing starts at 0

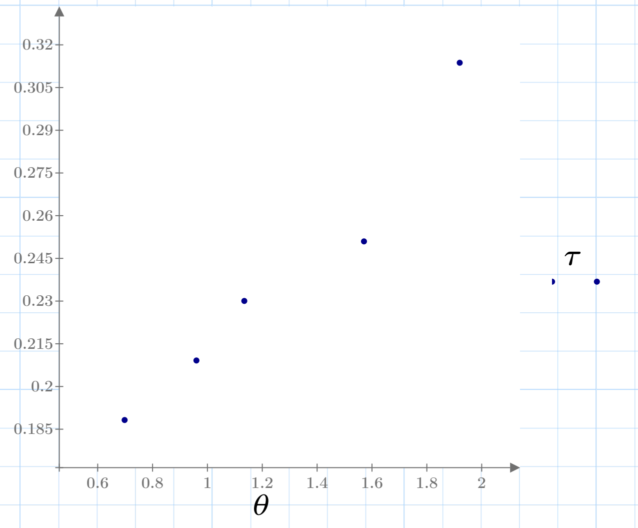
To the left is a dataset from Kau's NR-MFC 06.03's Example 1 where we are playing with mousetraps...

Let's use this case to explore basic linear regression theory and execution

First, let's plot these two fields against each other...

It clearly makes a nice linear relationship.

Let's now pull some basic stats...



$$\theta_{mean} := \text{mean}(\theta) = 1.257$$
$$s_\theta := \text{Stdev}(\theta) = 0.488$$
$$\tau_{mean} := \text{mean}(\tau) = 0.238$$
$$s_\tau := \text{Stdev}(\tau) = 0.048$$
$$N := \text{rows}(\theta) = 5$$

$$CI95_\theta := \text{qt}\left(1 - \frac{0.05}{2}, N-1\right) \cdot \frac{s_\theta}{\sqrt{N}} = 0.606$$

$$CI95_\tau := \text{qt}\left(1 - \frac{0.05}{2}, N-1\right) \cdot \frac{s_\tau}{\sqrt{N}} = 0.06$$

We will be creating a function to fit the dependent variable τ to independant variable θ

$$\tau_\theta(\theta) := a_0 + \boxed{a_1} \cdot \theta$$

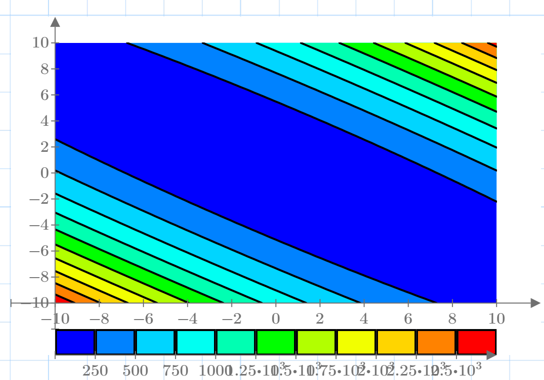
Remember the secret to linear regression. We want to minimize the error between our regression equation results and our original values of tau. We express it as a summed square of the residuals (SSR) or Sr

$$S_r(a_0, a_1) := \sum\left(\left(\left(a_0 + a_1 \cdot \theta\right) - \tau\right)^2\right)$$

Here we are plotting Sr vs a1 and a2 in a contour plot

Somewhere is the blue is the minimum Sr.

Since we went to college we can be clever and use some Calc-1-Fu. We want to find the place in the Sr(a1,a2) function where the gradient is zero



$$S_r(A1, A2)$$

Linear Regression Sand Box and Open Hand Problem

There are a few clever ways to do this in Matchad. If you take the derivative (slope or gradient) of the Sr(a1,a2) and solve it using the symbolic arrow...

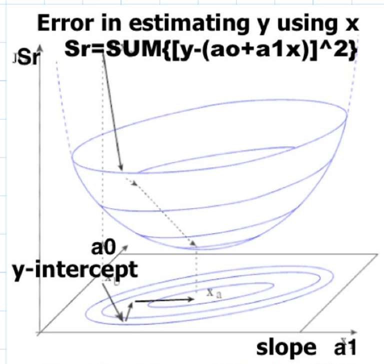
$$S_{r0}\left(A_0,A_1\right) := \frac{\mathrm{d}}{\mathrm{d}A_0} S_r\left(A_0,A_1\right) \rightarrow 10 \cdot A_0 + 12.56637 \cdot A_1 - 2.384172$$

$$S_{r1}\left(A_0,A_1\right) := \frac{\mathrm{d}}{\mathrm{d}A_1} S_r\left(A_0,A_1\right) \rightarrow 12.56637 \cdot A_0 + 17.698269108282 \cdot A_1 - 3.179275851496$$

So we have two simulaneous equations
(just like with Mrs Mercer in Junior High School Algebra Class)
And here we want to solve for the values of a1 and a0 when both derivatives (the total gradient of our "error bowl") is zero

Two equations. Two unknowns. We can solve this system!

In Mathcad there are a couple cool ways to solve these two equations for given that BOTH derivatives are equal to zero (i.e., where the Sr(a0,a1) function creates the bottom of the "bowl.")



from Bouttier and Courtier 1999

Here's one... (We'll have fun with more methods later in the course!)

$$A := \begin{bmatrix} S_{r0}\left(A_0,A_1\right) \\ S_{r1}\left(A_0,A_1\right) \end{bmatrix} \xrightarrow{solve\,,A_0\,,A_1} \begin{bmatrix} 0.11766514898834059588 & 0.096091433732779954845 \end{bmatrix}$$

Hypoethetically if this worked... Our slope and intercept would be..

y-Intercept: $A^{\langle 0 \rangle} = \left[1.177 \cdot 10^{-1}\right]$ Slope: $A^{\langle 1 \rangle} = \left[9.609 \cdot 10^{-2}\right]$

Let's double-check this against our available intrinsic functions in Matchad (and Excel) and also using the equations in Mathcad.

Since this is an open hand case and we are replicating the NR-MFC text we have an additonal point on which to bench our results!

$$\begin{aligned} k_1 &= \bar{T} - k_2\,\bar{\theta} \\ &= 2.3842\times10^{-1} - (9.6091\times10^{-2})(1.2566) \\ &= 1.1767\times10^{-1} \ \ \mathrm{N\text{-}m} \end{aligned}$$

y-Intercept: $A^{\langle 0 \rangle} = \left[1.177 \cdot 10^{-1}\right]$

$$\begin{aligned} k_2 &= \frac{n\sum_{i=1}^{5}\theta_i T_i - \sum_{i=1}^{5}\theta_i \sum_{i=1}^{5} T_i}{n\sum_{i=1}^{5}\theta_i^2 - \left(\sum_{i=1}^{5}\theta_i\right)^2} \\ &= \frac{5(1.5896) - (6.2831)(1.1921)}{5(8.8491) - (6.2831)^2} \\ &= 9.6091\times10^{-2} \ \mathrm{N\text{-}m/rad} \end{aligned}$$

Slope: $A^{\langle 1 \rangle} = \left[9.609 \cdot 10^{-2}\right]$

And now the rest is on you! Try these formulas by hand as well as the intrinsic functions...

Linear Regression Sand Box and Open Hand Problem

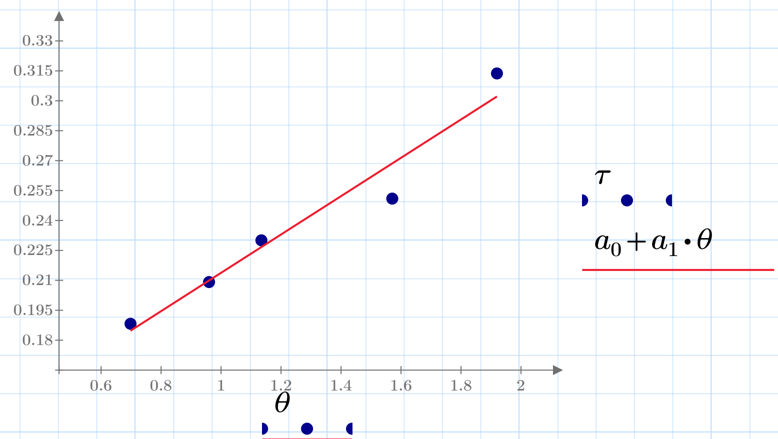
$$slope\_c := \frac{N \cdot \sum_{i=0}^{N-1} \left(\theta_i \, \tau_i\right) - \sum \theta \cdot \sum \tau}{N \cdot \sum \theta^2 - \left(\sum \theta\right)^2} = 0.096 \qquad yint\_c := \tau_{mean} - slope\_c \cdot \theta_{mean} = 0.118$$

$$a_1 := \text{slope}(\theta, \tau) = 0.096 \qquad a_0 := \text{intercept}(\theta, \tau) = 0.118$$

We have satisfactory benches with all methods!

Now let's examine the quality of our regression.
(Just by visual inspection you should expect a low RMSE and high correlations
Let's start by creating a new formula for our linear regression

$$\tau_\theta(\theta) := a_0 + a_1 \cdot \theta$$



The Total Sum of Squares (St or SST) is the baseline unskilled error and is the unskilled analog (i.e., equivalent) to the Sum of the Squred Residuals (Sr or SSR). This is basically the varaince formula of your dependant variable (torque) without dividing by N

$$S_t := \sum\left(\left(\tau - \tau_{mean}\right)^2\right) = 9.273 \cdot 10^{-3}$$

And our Sr (our skilled error) is...

$$S_r := \sum\left(\left(\tau - \tau_\theta(\theta)\right)^2\right) = 4.688 \cdot 10^{-4}$$

Clearly we have an improvement here. So let's articulate it with our error metrics.
Our RMSE and Standard Error of the Estimate are...

$$RMSE := \sqrt{\frac{S_r}{N}} = 9.683 \cdot 10^{-3} \qquad s_{\tau\theta} := \sqrt{\frac{S_r}{N-2}} = 1.25 \cdot 10^{-2}$$

And using a basic baseline for skill where the error for a perfect forecast skill is zero.
Remember that a good working definition of "skill" is how much better your method is to random chance or other unskilled method

$$skill := \frac{S_r - S_t}{0 - S_t} = 0.949 \qquad r2 := \frac{S_t - S_r}{S_t - 0} = 0.949$$

$$\text{corr}(\theta, \tau) = 0.974$$

$$\text{corr}(\theta, \tau)^2 = 0.949$$

Now try this with your Automobile Dataset